A Seamless and Efficient Transition Algorithm For Aerial Drone Multicasting

Wanqing Tu
School of Computer Science, The University of Auckland, New Zealand.
Email: w.tu@auckland.ac.nz

***Abstract*—Drone communications make use of line-of-sight coverage of drones to realise services that ground devices may not support. Many relevant applications such as video capture by drones and drone traffic management, require group communications between drones to efficiently disseminate data. In this paper, we study high-performance yet resource-efficient multicasting between drones that may change their locations in order to fulfill their missions. This is achieved by proposing novel trajectories for mobile drones to seamlessly transit, with controlled travel distances and traffic overheads, in a multicasting environment. The presented *efficient transition via trajectory adjustment* (ETTA) algorithm is developed based on our analysis of the condition that determines when a straight-line trajectory between the origin and destination of a drone is not seamless. The algorithm then proposes the trajectory adjustment schemes that form a new interference-controlled and travel-distance-controlled trajectory to replace an interrupted straight-line trajectory. Our NS2 simulation results demonstrate that ETTA, as compared to other mobile multicasts, can achieve guaranteed performance in a multicast with heavier traffic loads.**

I. Introduction

Many drone-related applications and services, e.g., aerial video capture by drones, air rescue assistance, surveillance operations, drone traffic management, drone safety monitoring, will benefit from group communications. Multicasting between drones is essential for these group applications, not only because it may deliver data with rich content to multiple receivers by using much less transmission resources but also because it allows critical information (e.g., drone safety alarms, drone traffic schedules) to be reliably received via multiple receivers. Drone multicasts can also help ground multicasts to relay information to members that are not within the line of sight of each other. In these group applications, drones often change their locations in order to capture information from different angles, adjust line of sight ranges between sky and ground, etc. This paper studies aerial multicasting between drones that may change their locations in order to fulfill their missions, supporting the development of relevant drone applications.

Mobile drone multicasting inherits the challenges faced by mobile multicasting on the ground: node transitions cause interrupted connections or increased interference. Conventional ground solutions enhance tree-based or mesh-based multicasting protocols [1-2] to avoid interruption or interference. They however introduce considerable traffic overheads or complex maintenance for networks and their devices, making them unsuitable for drones that often have energy constraints and computation limitations. More popular studies develop geographic multicasting [3-6]. In general, geographic multicasting arranges group members into different physical zones based on their locations. Multicasting trees are established to connect different zones and broadcast is used within each zone. Zone leaders manage members' joining or leaving which decreases the requirement to adjust multicasting trees between zones, greatly reducing the incidence of interrupted mobile connections and the associated interference and overheads. However, broadcasting unnecessarily spreads data to nodes that do not belong to a group, wasting wireless bandwidth and device energy. Moreover, it is complicated to plan or form zones based on nodes' physical locations.

For drone-related multicasting, the literature mainly focuses on drone-to-earth applications (e.g., [7-8]) in which a drone disseminates data to a set of ground users/devices. Multicasting between drones is rarely studied. The resource limitations of drones and their wireless connections require handover operations to make light use of resources while guaranteeing seamless transitions. Therefore, drone transitions with low traffic overhead, controlled energy consumption, and simple computation tasks form our design target. Our transition takes advantage of the existence of multiple forwarders in a multicasting as well as an obstacle-free aerial communication environment to achieve the expected transition performance. In our system, straight-line trajectories are adopted with priority to transfer drones, incurring short travel distances and low traffic overheads and hence benefitting resource efficiency. However, straight-line trajectories may not always be seamless. We hence design a new algorithm - efficient transition via trajectory adjustment (ETTA) to guide drones' movement with controlled resource consumption. In detail, our contributions include the following results.

- Trajectory adjustment condition. We analyse the condition when a straight-line trajectory is not fully covered by the multicasting system if a drone transits between forwarders that have overlapping coverage. The implementation of such condition should be a fast yet resource-efficient process as it mostly requires to calculate Euclidean distance based on the 3-dimensional Pythagorean theorem.
- Efficient trajectory adjustment schemes. The schemes propose new drone trajectories to replace interrupted straight-line trajectories when drones transit between forwarders with or without overlapping coverage. When drones move between forwarders with overlapping coverage, our new trajectories are established to limit additional travel distances over those of the original straight-line trajectories, balancing the tradeoff between enabling

fast transitions and controlling resource utilisation. For transitions between forwarders without overlapping coverage, the tradeoff is balanced by employing a minimal number of multicasting forwarders to seamlessly cover drones transiting between origins and destinations.
- The ETTA algorithm. It systematically combines the trajectory adjustment condition and the efficient trajectory adjustment scheme, supporting efficient yet seamless drone transitions in aerial multicasting.

Finally, we use NS2 simulations to evaluate our ETTA. We observe the average multicast delays and the average multicast throughput in different multicasting networks. The results show that, as compared to existing studies, ETTA may admit 68% more traffic load while guaranteeing the multicasting performance for both mobile and stable drone receivers.

II. Related Studies

Wireless multicast with static group members focuses on improving complex interference and limited wireless bandwidth. Early strategies avoid interference by utilising non-overlapping channels between nearby nodes (e.g., [9]) or by hopping nodes between different channels (e.g., [10]). Transmission scheduling is another well studied strategy that efficiently utilises channel resources to gain more transmission opportunities. Studies have scheduled transmission rates (e.g., [11]), flow transmissions (e.g., [12]), etc. to enable a channel to accommodate more multicasts or to extend multicasts' coverage. External resources, such as licensed RF bands (e.g., [13]) or wired network links (e.g., [14-15]), are also exploited for additional bandwidth.

For mobile multicasting, many studies concern the reliable performance received by mobile members. In [16], a tree multicast is designed that assigns an ID to each multicasting node. Flows are forwarded in order of IDs. Interrupted connectivity is repaired by referring to the sparseness between IDs. The core-assisted mesh protocol [1] builds a shared multicast mesh to maintain group connectivity when network routers move frequently. It reverses the shortest unicast paths to form multicasting paths on the shared mesh, supporting loop-free packet forwarding. Tree- or mesh-based mobile multicasting often requires complex operations to maintain connectivity, generating considerable overheads to bandwidth-limited mobile networks. Geographic multicasting (e.g., [3]) improves this drawback by dividing group members into different zones. These zones are connected via a multicasting. This multicasting structure does not change with nodes' mobility because their movements do not cause zone movement. Within each zone, data is delivered via greedy forwarding. The geographic multicasting protocol in [4] computes a Steiner tree to connect zones. It carries concurrent multicasts for a higher delivery ratio, resulting in scalable delay performance even when network sizes increase. The work in [17] designs a virtual-zone-based structure to manage group members. With the position information of multicasting nodes, it constructs a zone-based bidirectional tree. The protocol uses zone depth to optimise tree structures and integrates nodes' location information with group member management to enhance multicasting efficiency.

For drone-related multicasting, in [7], drone-to-earth multicast transmissions are developed by using filter bank multicarrier. The proposal designs filter bank multicarrier with offset quadrature amplitude modulation (offset-QAM) and hermite polynomial-based prototype filtering, helping to manage the tradeoff between performance and spectral efficiency. In [8], drone trajectories are designed theoretically to minimise mission completion time while ensuring each ground terminal to recover the file with a high probability. In general, while multicasting between drones is important to support many emerging, it is however rarely studied in literature.

III. Efficient Transition Via Trajectory Adjustment (ETTA)

In our system, by an existing wireless multicasting algorithm (e.g., [14-15,17]), drones form a multicasting architecture. Drone transitions on this multicasting architecture can be between overlapping forwarders (i.e., two forwarders with overlapping coverage) or non-overlapping forwarders (i.e., forwarders without overlapping coverage). In this section, we study efficient and seamless drone transitions for both transition scenarios. We then present the *efficient transition via trajectory adjustment* (ETTA) algorithm that systematically integrates our theoretical analysis and trajectory adjustment schemes to provide travel-distance-controlled and interference-controlled trajectories, with controlled traffic overheads, for mobile drones in aerial multicasting.

A. Drone Transitions Between Overlapping Forwarders

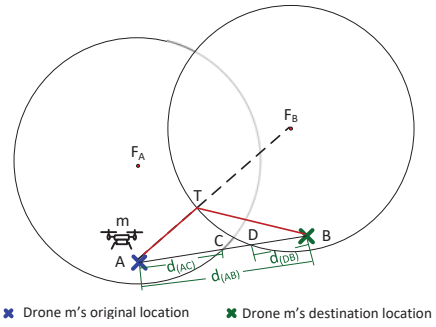

Fig. 1. Deriving the trajectory adjustment condition.

As mentioned, straight-line trajectories support fast drone transitions and use resources efficiently. However, they may not always be seamless. We use Fig. 1 to illustrate how to determine whether a straight-line trajectory is seamless or not when a mobile drone m transits between two overlapping forwarders. In the figure, if m moves from A to B, let m's original and destination forwarders be F_A and F_B, and the intersections of the straight-line trajectory with the coverage edges of F_A and F_B be C and D respectively. Denote the distances between A and B, A and C, and B and D as $d_{(AB)}$, $d_{(AC)}$, and $d_{(BD)}$ respectively. In addition, drones in our system communicate via omnidirectional antennas in an obstacle-free space (i.e., the sky). We hence assume that a drone's transmission range is a sphere. Without loss of

generality, we let the radius of the coverage sphere be r. Theorem 1 gives the trajectory adjustment condition.

Theorem 1. *For a drone m moving between two overlapping forwarders (shown in Fig. 1), the straight-line trajectory from its origin A to its destination B is seamless if one of the following conditions meets: 1) $d_{(AB)} \leq d_{(AC)} + d_{(DB)}$, or 2) when $d_{(AB)} > d_{(AC)} + d_{(DB)}$, there exists a forwarder in the multicasting whose distances to C and D are both $\leq r$. Otherwise, the straight-line trajectory needs to be adjusted.*

Proof. We prove Theorem 1 by contradiction. When $d_{(AB)} \leq d_{(AC)} + d_{(DB)}$, suppose the straight line $A \rightarrow B$ is not seamless. Then, some part(s) of the straight-line trajectory is(are) not covered by F_A and F_B. Let the length of the uncovered part(s) be $l > 0$. We have $d_{(AC)} + l + d_{(DB)} = d_{(AB)} \Rightarrow l = d_{(AB)} - d_{(AC)} - d_{(DB)}$. Since $l > 0$, we have $d_{(AB)} > d_{(AC)} + d_{(DB)}$. This contradicts $d_{(AB)} \leq d_{(AC)} + d_{(DB)}$. Therefore, when $d_{(AB)} \leq d_{(AC)} + d_{(DB)}$, the straight-line trajectory does not need to be adjusted.

When $d_{(AB)} > d_{(AC)} + d_{(DB)}$, suppose there exists a multicasting forwarder f whose distances to C and D are both $\leq r$. If the straight-line trajectory is not seamless, there is at least a point between C and D whose Euclidean distance to f is $> r$. This makes that $C \rightarrow D$ is not a straight line because the two ends C and D are both within the distance of r to f, contradicting the fact that $A \rightarrow B$ is a straight line. Q.E.D

The implementation of Theorem 1 requires knowledge of the Euclidean coordinates of C and D. We denote the cooridnates as (x_C, y_C, z_C) and (x_D, y_D, z_D) respectively. As C is on the edge of F_A's transmission range, we have

$$(x_C - x_{F_A})^2 + (y_C - y_{F_A})^2 + (z_C - z_{F_A})^2 = r^2. \quad (1)$$

Also, C is on the straight-line trajectory $A \rightarrow B$, i.e.,

$$\begin{cases} x_C = x_A + t(x_B - x_A), \\ y_C = y_A + t(y_B - y_A), \\ z_C = z_A + t(z_B - z_A). \end{cases} \quad (2)$$

Inputting (2) into (1), by solving the quadratic equation for t, typically two distinct values of t will be obtained which define two distinct points. The point closer to F_B is C. Similarly, the coordinates of D can be obtained. With the coordinates of C and D, by Theorem 1, if $A \rightarrow B$ is seamless, m transits via this straight-line trajectory. Otherwise, a new trajectory is formed as below.

When proposing a new trajectory, we try to control traffic overheads and m's travel distances with computation of low complexity, allowing fast transitions with efficient use of resources (e.g., energy, bandwidth). The idea is to employ a location (denoted as T), within the overlap of transmission ranges of F_A and F_B, to form a transition path $A \rightarrow T \rightarrow B$ inside the combined coverage of F_A and F_B. Ideally, T should minimise the extra travel distance exceeding that of the straight-line trajectory. Such a location is achievable by an existing algorithm (e.g., [19]) to seek a point on the surface of the overlapping area that has the shortest distance to the straight line $A \rightarrow B$. However, this potentially increases computation delays and its energy consumption. Therefore, we use the closest intersection between the line $A \rightarrow F_B$ and F_B's coverage edge to the straight-line trajectory as T (illustrated in Fig. 1). By the line function between A and F_B, T's coordinates can be formulised as below,

$$\begin{cases} x_T = t(x_{F_B} - x_A) + x_A, \\ y_T = t(y_{F_B} - y_A) + y_A, \\ z_T = t(z_{F_B} - z_A) + z_A. \end{cases} \quad (3)$$

Furthermore, as T is on the edge of F_B's coverage, we have $(x_T - x_{F_B})^2 + (y_T - y_{F_B})^2 + (z_T - z_{F_B})^2 = r^2$. Combining this equation with (3), we can derive t and hence T's coordinates. The new trajectory $A \rightarrow T \rightarrow B$ is shown by the red lines in Fig. 1.

B. Drone Transitions Between Non-Overlapping Forwarders

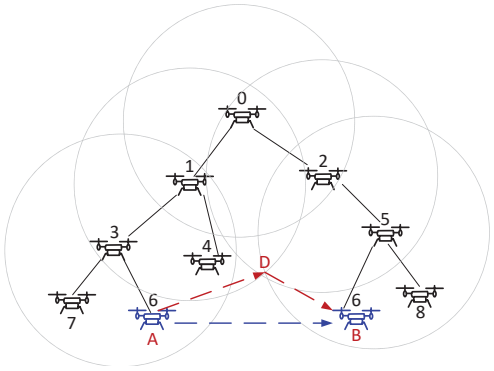

Fig. 2. An example of ETTA's multicasting architecture and forming an ETTA trajectory.

Recall that, drones in our system form a multi-hop multicasting architecture between them by an existing wireless multicasting algorithm (e.g., [14-15,17]). Fig. 2 illustrates an example of such a multi-hop drone multicasting architecture. When transiting a drone (e.g., drone 6 in Fig. 2) between non-overlapping forwarders, in order to form a seamless trajectory with controlled travel distance to replace an interrupted straight-line trajectory (e.g., the blue dotted line in Fig. 2), our idea is to select a minimal number of multicasting forwarders that overlap one by one to provide coverage along the transition path. In detail, m generates an overlapping graph to represent how multicasting forwarders' coverage overlaps. Multicasting forwarders are nodes on this graph. If two forwarders are overlapping, an edge between nodes representing the two forwarders is added to the graph. Fig. 3 shows the overlapping graph of the multicasting tree in Fig. 2.

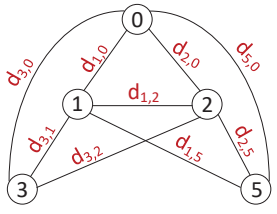

Fig. 3. The overlapping graph of the multicasting tree in Fig. 2.

On this overlapping graph, each edge has a weight. Denote the weight of edge i ($i \in [0, e-1]$) connecting two overlapping forwarders (say f' and f'') as ω_i, where e is the total number of edges in the graph. For obtaining a short-delay trajectory, ω_i is the Euclidean distance between f' and f'', namely,

$$\omega_i = d_{f',f''} = \sqrt{(x_{f'} - x_{f''})^2 + (y_{f'} - y_{f''})^2 + (z_{f'} - z_{f''})^2}, \quad (4)$$

where $(x_{f'}, y_{f'}, z_{f'})$ and $(x_{f''}, y_{f''}, z_{f''})$ are the coordinates of f' and f'' respectively. In Fig. 3, the red distance symbols are edge weights achieved by (4). Via this weighted overlapping graph, by employing existing algorithms (e.g., Dijkstra's algorithm, the A* search algorithm), m searches the path that connects its original forwarder to its destination forwarder with the lowest weight value.

Based on the selected path, m starts forming a seamless trajectory. Excluding the original and destination forwarders, all other forwarders on the selected path are referred as m's trajectory forwarders. Suppose there are n trajectory forwarders with the ith ($i \in [0, n-1]$) denoted TF_i. m calculates the intersections between the coverage edges of F_A and TF_0, the coverage edges of TF_j and $TF_{(j+1)}$ ($j \in [0, n-2]$), and the coverage edges of $TF_{(n-1)}$ and F_B. Typically two distinct intersections will be obtained for each pair of consecutive forwarders on the selected path. The one closer to m's destination B, called an eligible intersection (EI), is employed to participate in forming a part of m's trajectory. In detail, with the first EI, m employs Theorem 1 to check the seamlessness of the straight line between m's original A and this EI. If seamless, the straight line forms part of m's trajectory. If not, m employs the scheme in Section III. A to locate T which helps to form the first part ($A \rightarrow T \rightarrow$ the first EI) of the seamless trajectory. For the remaining part of the trajectory, m uses the straight lines connecting consecutive EIs. This is because any two consecutive EIs are covered by the same trajectory forwarder, ensuring that the straight line between them is seamless. For the last part of m's trajectory, the straight line between the EI (selected based on the coverage edges of $TF_{(n-1)}$ and B) and B is employed because both the EI and B are covered by F_B.

We use an example in Fig. 2 to illustrate the above trajectory formation. Following the graph in Fig. 3, suppose drone 6 selects drones 3, 1, & 5 to support its transition. Drone 6 calculates the EI (denoted as D in Fig. 2) between the coverage edges of drones 1 & 5. By Theorem 1, $A \rightarrow D$ is seamless and hence is included as part of the trajectory. Now, as drone 1 overlaps with the destination forwarder drone 5 and drone 5 covers both D and B, $D \rightarrow B$ becomes the remaining part of the trajectory. The red dotted arrow lines show the trajectory.

C. The ETTA Algorithm

When establishing a multicasting architecture between drones, selected forwarders exchange location information[1], and then calculates and exchanges their Euclidean distances to each other. Then, the ETTA algorithm combining our studies in previous subsections is employed to transit mobile drones in a seamless and resource-efficient manner.

[1]The location information may be obtainable for example via a GPS receiver. Research studies (e.g., [21]) also proposed good schemes to locate nodes in mobile ad-hoc networks.

Algorithm 1 Efficient Transition via Trajectory Adjustment

Input: Mobile drone m, m's origin (A) and destination (B), m's origin and destination forwarders F_A and F_B;
Output:m's ETTA transition trajectory from A to B.

1. m checks whether F_A and F_B are overlapping or not;
2. If overlapping, by Theorem 1, m checks whether the straight-line trajectory is seamless or not;
3. If so, m transits via $A \rightarrow B$ directly; Exit.
4. If not, m decides T to form a new seamless trajectory $A \rightarrow T \rightarrow B$ to transit; Exit.
5. If non-overlapping,
6. m generates an overlapping graph for multicasting forwarders; m assigns weights to edges on the graph by (4);
7. m employs Dijkstra's or A* algorithm to find a path with the minimum weight value; suppose n trajectory forwarders on the path;
8. m calculates the EI between the edges of the coverage of TF_0 and TF_1;
9. If the trajectory ($A \rightarrow$ this current EI) is seamless based on Theorem 1, it becomes part of m's trajectory;
10. Otherwise, m forms ($A \rightarrow T \rightarrow$ this current EI);
11. $i = 1$;
12. While $i < n - 1$
13. m calculates the EI between the coverage edges of TF_i and $TF_{(i+1)}$; the straight line between the last EI and this EI becomes part of m's trajectory; $i = i + 1$;
14. m calculates the EI between the coverage edges of $TF_{(n-1)}$ and F_B; the straight line from the EI to B is the last part of m's trajectory; Exit.

IV. Simulation Evaluations

TABLE I
Simulation Parameters

Parameters	Values	Parameters	Values
Frequency	2.4GHz	Propagation model	Free space
Dimensions	3D	Transmission power	15dBm
Number of channels	1	Wireless channel data rate	54Mbps
Simulation time	200s	MAC protocol	802.11
Antenna	Omnidirectional antenna	Receive threshold	-80dBm

We conduct simulation studies in NS2.35 [18] to compare three related multicasting schemes with our ETTA when they handle mobile group members: LCRT [14-15] employs the minimum number of forwarders to multicast data, reducing interference but not supporting any drone mobility; T-LCRT enhances LCRT by selecting drones on the multicasting tree to support transitions. Drone receivers may forward data to mobile drones if they are nearby; EGMP [17], is a geographic multicast, grouping drones into zones which are connected

via a bi-directional tree. In our simulations, as LCRT controls interference well, ETTA builds a LCRT tree to connect drones. There are a few other recent studies on drone-related multicasting (e.g., [7-8]). They mostly focus on drone-to-earth single-hop data multicasting and hence we do not compare ETTA with them as our study explores multi-hop drone-to-drone multicasting.

Table I lists common settings used in our simulations. We observe the average multicast delays (AMD), the average multicast throughput (AMT), and the average mobile throughput (AMoT). AMD is calculated by $AMD = \frac{AD_i}{n}$, and AMT is calculated by $AMT = \frac{AT_i}{n}$, where n is the total number of drone receivers, $i \in [0, n-1]$, $j \in [0, m-1]$, and AD_i and AT_i are the average data delay and the average data throughput at ith drone receiver. Our results plotted are the mean values of 20 simulation runs.

A. Evaluation of Small-Group Mobile Multicasting

We first conduct a small-group simulation with 9 drones. Two mobile drones exist: the first one moves a distance of 102.6 meters at a speed of 10m/s, and the second one moves a distance of 76 meters at a speed of 20m/s. Fig. 4 shows the AMD performance. LCRT and ETTA achieve shorter AMDs than T-LCRT and EGMP do. This is because T-LCRT and EGMP employ nodes that are not forwarders on the multicasting structure to support transitions, while ETTA makes use of multicasting forwarders to handover mobile drones and LCRT does not implement any handover process. The employment of transition forwarders that are not on the multicasting structure generates extra traffic to the system, prolonging the multicasting delays of T-LCRT and EGMP. Furthermore, T-LCRT issues control traffic to the system in order to determine suitable transition forwarders, worsening T-LCRT's AMDs as compared to EGMP. Both LCRT and ETTA achieve AMDs under 150ms in this simulation. The slight AMD difference is because they calculate AMDs based on different packets: when calculating AMDs, LCRT does not consider those packets dropped during mobile transition while ETTA counts all transmitted packets.

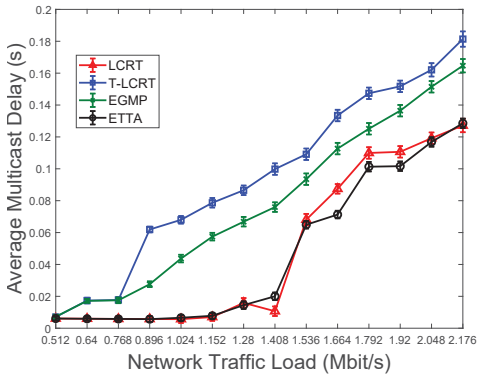

Fig. 4. Comparison of AMDs in the small-group simulation.

Fig. 5 plots AMT performance. ETTA achieves the highest AMT by transiting mobile drones via trajectories fully covered by the multicasting tree. Hence, there is no extra node other than multicasting forwarders generating data traffic in the system. Also, the ETTA trajectory is planned using multicasting forwarders' coordinates which are obtained when establishing the multicasting tree, generating little control traffic in the system. LCRT has the lowest AMT because mobile drones do not receive data while moving. EGMP and T-LCRT both employ transition forwarders to support the transition, allowing them to achieve higher AMTs than LCRT. In addition, EGMP employs transition forwarders without changing its multicast architecture and EGMP transition forwarders can provide timely transitions, helping to achieve a higher AMT than T-LCRT.

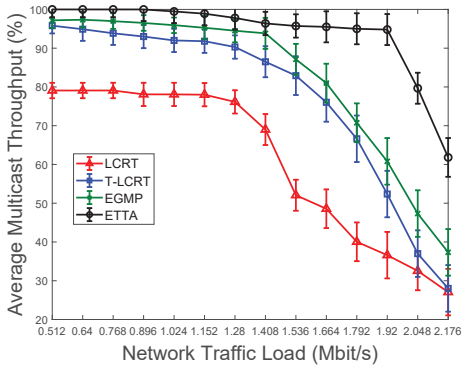

Fig. 5. Comparison of AMTs in the small-group simulation.

Overall, in this simulation, ETTA achieves better AMDs and AMTs than other compared schemes by asking mobile drones to travel an average of 20 additional meters. A controlled travel distance helps to reduce energy consumed during drone transitions.

B. Evaluation of Large-Group Mobile Multicast

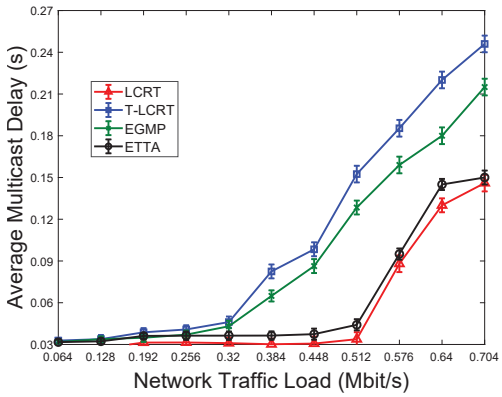

Fig. 6. Comparison of AMDs in the large-group simulation.

The large-group simulation has the 150 drones distributed so that each transmission range has 10 drones. There are 20 mobile drones, randomly selected by the simulation, with origins and destinations covered either by overlapping forwarders, short-distance non-overlapping forwarders, or long-distance

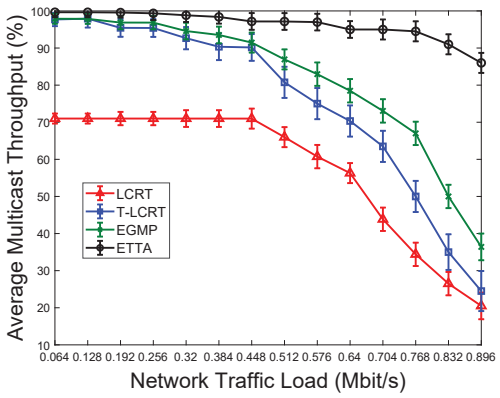

Fig. 7. Comparison of AMTs in the large-group simulation.

non-overlapping forwarders. These mobile drones start transiting at different times and move at different speeds ranging from 10m/s to 25m/s. We evaluate the four schemes when the network traffic load varies from 64Kbit/s to 896Kbit/s. Based on Fig. 6, the four schemes yield similar relative results for AMDs in the large-group simulation as AMDs from the small-group simulation. Similar reasons for the results in Fig. 4 can explain the results plotted in Fig. 6.

In Fig. 7, ETTA achieves higher AMT performance than other protocols. As compared to the AMT from the small-group simulation (in Fig. 5), although the relative results are similar, ETTA outperforms other protocols by a wider margin in the large-group simulation. ETTA achieves good AMT (90% or above) when the traffic load is $\leq$840Mb/s, while EGMP and T-LCRT achieve good AMT when the traffic load is less than 500Mb/s and 448Mb/s respectively. In another words, ETTA carries 68% or 87.5% more traffic with guaranteed AMTs than EGMP and T-LCRT. This is because in the large-group simulations, EGMP and T-LCRT require more complicated procedures or take more time to find transition forwarders. More transition forwarders also issue more extra traffic to the system.

Overall, in this simulation, ETTA achieves better AMDs and AMTs than other compared schemes by asking mobile drones to travel an average of around 85 additional meters. A controlled travel distance helps to reduce energy consumed during drone transitions.

V. Conclusion

In this paper, we studied drone multicasting in order to enable high-performance group communications between drones. Our development focused on how to seamlessly transit mobile drones in a resource-efficient manner, given the resource limitations experienced by drones and their wireless connections. A new algorithm, ETTA, was proposed that takes advantage of the obstacle-free aerial communication environment to establish straight-line trajectories for mobile drones. As straight-line trajectories may not always be seamless, we theoretically presented the trajectory adjustment condition by which the ETTA algorithm can determine the seamlessness of a straight-line trajectory. To replace an interrupted straight-line trajectory, we proposed new schemes to form a distance-controlled trajectory with forwarders already on the multicasting tree. As such, the ETTA algorithm allows fast drone transitions while controlling the traffic overheads issued to the network. Our simulation results proved that ETTA delivers multicast data with acceptable performance when the multicasting system carries 68% more traffic than compared mobile multicasting protocols.

Acknowledgement

This research is supported by InternetNZ research grant GR000262019/20 and UoA FRDF grant 3719447.

References

[1] J. Garcia-Luna-Aceves, and E. Madruga. The Core-assisted Mesh Protocol. IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pages 1380-1394, 2006.

[2] W. Tu, C. Sreenan, C. Chou, A. Misra, and S. Jha. Resource-Aware Video Multicasting via Access Gateways in Wireless Mesh Network. IEEE Transactions on Mobile Computing, vol. 11, issue 6, pages 881-895, 2012.

[3] S. Wu, and K. Candan. GMP: Distributed geographic multicast routing in wireless sensor networks. In Proc. of the 26th IEEE ICDCS, pages 1-9, Portugal, 2006.

[4] J. Adamek, M. Nesterenko, and J. Robinson. Concurrent Geometric Multicasting. In Proc. of The 19th ICDCN, 9:1 - 9:10, 2018.

[5] X. Xiang, X. Wang, and Y. Yang. Supporting Efficient and Scalable Multicasting Over Mobile Ad Hoc Networks. IEEE Transactions on Mobile Computing, vol. 10, issue 4, pages 544-559, 2010.

[6] H. Hussen, S. Choi, J. Park, and J. Kim. Predictive Geographic Multicast Routing Protocol in Flying Ad Hoc Networks. International Journal of Distributed Sensor Networks, pages 1-20, 2019.

[7] C. Sacchi, F. Granelli, and T. Rahman. Filter-Bank Multicarrier Transmission for Drone-to-Earth Multicast Applications. In Proc. of IEEE ISBMSB, pages 1-6, Spain, 2018.

[8] Y. Zeng, X. Xu, and R. Zhang. Trajectory Design for Completion Time Minimization in UAV-Enabled Multicasting. IEEE Transactions on Wireless Communications, vol. 17, issue 4, pages 2233-2246, 2018.

[9] R. Raniwala, and T. Chiueh. Architecture And Algorithms For An IEEE 802.11 Based Multi-Channel Wireless Mesh Network. In Proc. of IEEE INFOCOM, pages 1-12, 2005.

[10] P. Bahl, R. Chandra, and J. Dunagan, SSCH: Slotted Seeded Channel Hopping For Capacity Improvement In IEEE 802.11 Ad-Hoc Wireless Networks. In Proc. of MOBICOM, pages 1-15, 2004.

[11] W. Tu. Efficient Wireless Multimedia Multicast in Multi-rate Multi-channel Mesh Network. IEEE Transactions on Signal and Information Processing over Networks, vol. 2, issue 3, pages 376-390, 2016.

[12] W. Tu. Efficent Resource Utilization for Multi-flow Wireless Multicasting Transmissions. IEEE Journal on Selected Areas in Communications, 30(7): 1246-1258, 2012.

[13] F. Hou, Z. Chen, J. Huang, Z. Li, and A. Katsaggelos. Multimedia multicast service provisioning in cognitive radio networks. In Proc. of IWCMC, pages 1175-1180, 2013.

[14] W. Tu, C. Sreenan, S. Jha, and Q. Zhang. Multi-source Video Multicast in Internet-connected Wireless Mesh Networks. IEEE Transactions on Mobile Computing, vol. 16, issue 12, pages 343-3444, 2017.

[15] W. Tu, C. Sreenan, C. Chou, A. Misra, and S. Jha. Resource-Aware Video Multicasting via Access Gateways in Wireless Mesh Network. IEEE Transactions on Mobile Computing, vol. 11, issue 6, pages 881-895, 2012.

[16] C. Wu, and Y. Tay. AMRIS: A Multicast Protocol for Ad Hoc Wireless Networks. In Proc. of IEEE MILCOM, pages 25-29, USA, 1999.

[17] X. Xiang, X. Wang, and Y. Yang. Supporting Efficient and Scalable Multicasting Over Mobile Ad Hoc Networks. IEEE Transactions on Mobile Computing, vol. 10, issue 4, pages 544-559, 2010.

[18] https://www.isi.edu/nsnam/ns/.

[19] David Eberly. Distance to Circles in 3D. Geometric Tools, pages 1-14, 2019.

[20] W. Tu, C. Sreenan, C. Chou, A. Misra, and S. Jha. Resource-Aware Video Multicasting via Access Gateways in Wireless Mesh Networks. In Proc. of IEEE ICNP, pages 43-52, USA, 2008.

[21] S. Capkun, M. Hamdi, and J. Hubaux. GPS-Free Positioning in Mobile Ad-hoc Networks. Cluster Computing 5(2), pages 1-15, 2002.